

Denoising Diffusion Models for Plug-and-Play Image Restoration

Yuanzhi Zhu

Supervisors: Dr. Kai Zhang, Jingyun Liang, Jiezhang Cao
Principal Investigator: Prof. Luv Van Gool



26/Jan/2023

Content

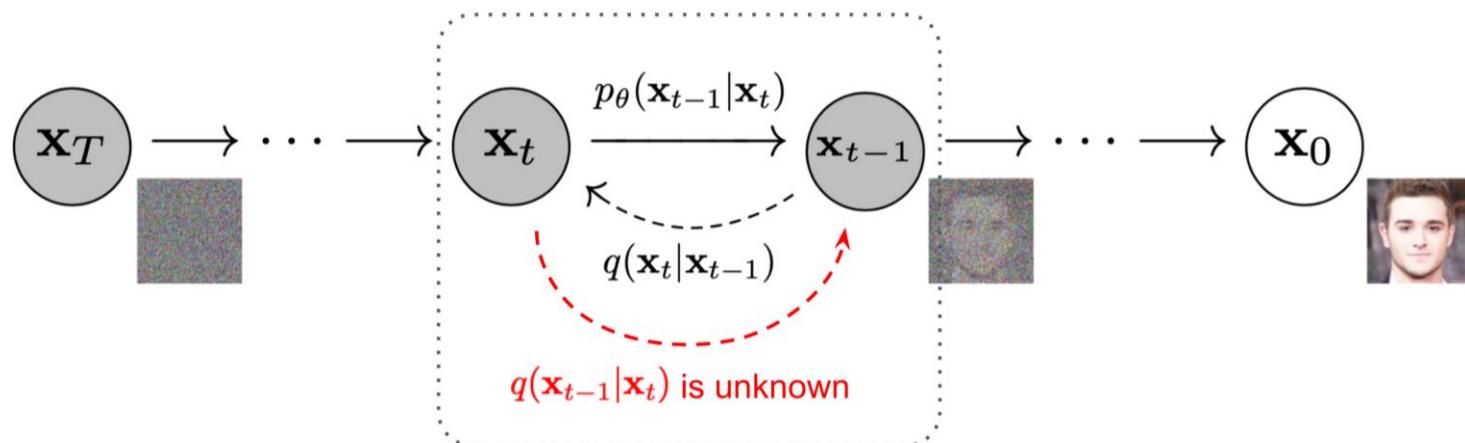
- Preliminaries
- Methods
- Results

DDPM: Denoising Diffusion Probabilistic Models*

True data dist. : $x_0 \sim q(x_0)$

Forward process: $q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1})$ Markov assumption

Reverse process: $p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t)$



DDPM: Denoising Diffusion Probabilistic Models

Forward Diffusion Process

$$q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1})$$

Each Step

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \frac{\sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}}{\text{norm invariant}}) \quad \text{or} \quad x_t = \sqrt{1 - \beta_t}x_{t-1} + \sqrt{\beta_t}\epsilon_{t-1}$$

noise schedule β_t controls the diffusion process

For arbitrary t

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad \alpha_t = 1 - \beta_t \text{ and } \bar{\alpha}_t = \prod_{i=1}^T \alpha_i$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \cdot z_t$$

DDPM: Denoising Diffusion Probabilistic Models

Reverse Diffusion Process

if β_t is small enough, $q(x_{t-1}|x_t)$ will also be Gaussian

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t))$$

Reverse when condition on x_0

$$q(x_{t-1}|x_t, x_0) = \boxed{q(x_t|x_{t-1}, x_0) \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}} \quad \text{all three are forward processes}$$

$$\longrightarrow q(x_{t-1}|x_t, x_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\boldsymbol{\beta}}_t \mathbf{I})$$

$$\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0 = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{z}_t \right)$$

model \mathbf{z}_t with NN 😊

DDPM: Denoising Diffusion Probabilistic Models

Negative Log Likelihood to Variational Lower Bound

$$-\log p_\theta(\mathbf{x}_0) \implies \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{1:T} | \mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \implies \sum_{t=2}^T \underbrace{D_{\text{KL}}[q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)]}_{L_t}$$

Known

Explicit parameterization

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t)) \quad \begin{cases} \boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_\theta(x_t, t)) \\ \boldsymbol{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathbf{I} \end{cases}$$

two options \dagger

$$\sigma^2 = \begin{cases} \beta_t \\ \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t \end{cases}$$

Model The Noise (Residual)

Denoiser

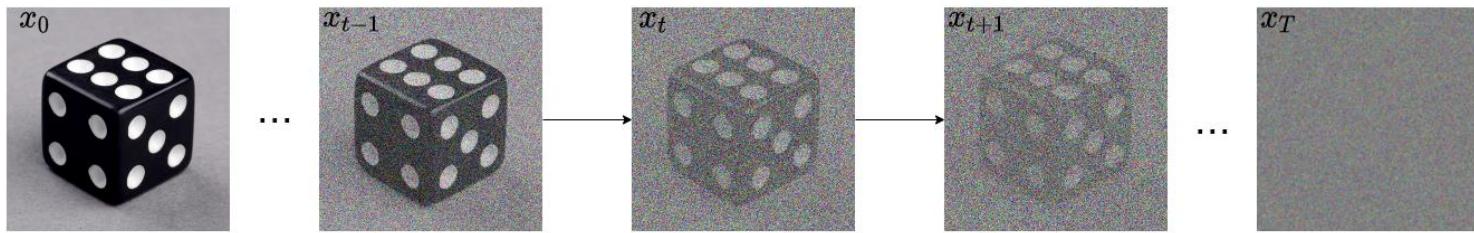
$$L_t^{\text{simple}} = \mathbb{E}_{x_0, z_t} \left[\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}z_t, t)\|^2 \right]$$

$$\hat{\mathbf{x}}_0^{(t)}(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \mathbf{z}_\theta^{(t)}(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$$

\dagger Covariance has analytical optimal form ([Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models](#))

DDPM: Denoising Diffusion Probabilistic Models

$$x_0 \sim q(x_0) \quad \xrightarrow{\text{red arrow}} \quad q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \quad \xrightarrow{\text{red arrow}} \quad q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1})$$



$$p_\theta(x_0) = \int p(x_T) \prod_{i=1}^T p_\theta(x_{t-1}|x_t) dx_{1:T} \quad \xleftarrow{\text{blue arrow}} \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad \xleftarrow{\text{blue arrow}} \quad x_T \sim \mathcal{N}(0, I)$$

Algorithm 1 Training

- 1: **repeat**
- 2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
- 3: $t \sim \text{Uniform}(\{1, \dots, T\})$
- 4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 5: Take gradient descent step on

$$\nabla_\theta \|\boldsymbol{\epsilon} - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$$
- 6: **until** converged

Algorithm 2 Sampling

- 1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 2: **for** $t = T, \dots, 1$ **do**
- 3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
- 4:
$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{z}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$$
- 5: **end for**
- 6: **return** \mathbf{x}_0

DDIM: Denoising Diffusion Implicit Models*

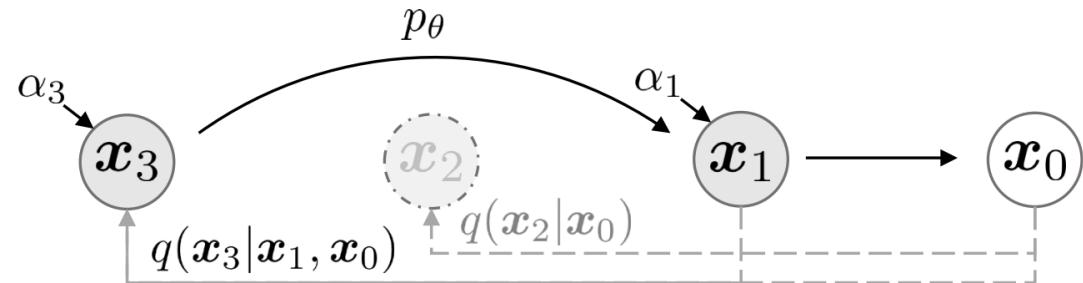
Reverse Process: deterministic given x_t, x_0

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1 - \bar{\alpha}_{t-1}} z_{t-1}$$

Sampling:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\underbrace{\frac{x_t - \sqrt{1 - \bar{\alpha}_t} z_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}}}_{\text{"predicted } x_0\text{"}} \right) + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1}} \cdot z_\theta^{(t)}(x_t)}_{\text{"direction pointing to } x_t\text{"}}$$

Accelerated Generation Processes



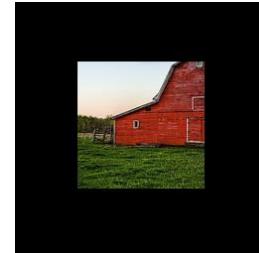
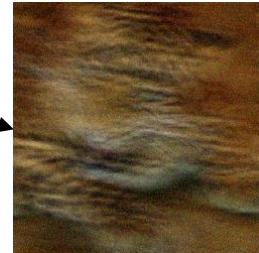
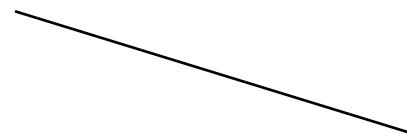
Content

- Preliminaries
- Methods
- Results

Plug-and-Play Image Restoration

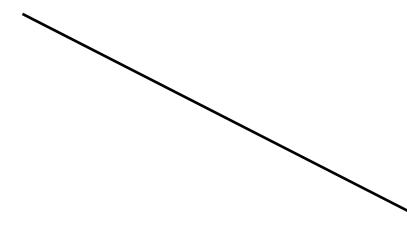
Measurement (degradation model):

$$\mathbf{y} = \mathcal{H}\mathbf{x} + \mathbf{n}$$



Reconstruction (Bayes' theorem):

$$\begin{aligned}\hat{\mathbf{x}} &\sim p(\mathbf{x}|\mathbf{y}) \\ &\propto p(\mathbf{x})p(\mathbf{y}|\mathbf{x})\end{aligned}$$



Maximum A Posteriori (MAP) estimation:

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x})$$

Plug-and-Play Image Restoration

Substitute degradation model $\mathbf{y} = \mathcal{H}\mathbf{x} + \mathbf{n}$: $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2\sigma_n^2} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda \mathcal{P}(\mathbf{x})$

$\frac{\text{data term}}{\text{prior term}}$

Introduce auxiliary variable \mathbf{z} :

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda \mathcal{P}(\mathbf{z}) \quad s.t. \quad \mathbf{z} = \mathbf{x}$$

Lagrange multiplier:

$$\mathcal{L}_\mu(\mathbf{x}, \mathbf{z}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \lambda \mathcal{P}(\mathbf{z}) + \frac{\mu}{2} \|\mathbf{z} - \mathbf{x}\|^2$$

Half Quadratic Splitting (**HQS**) algorithm:

$$\begin{cases} \mathbf{z}_k = \arg \min_{\mathbf{z}} \underbrace{\frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{z} - \mathbf{x}_k\|^2}_{\text{consistence}} + \underbrace{\mathcal{P}(\mathbf{z})}_{\text{prior}} & \text{Prior} \\ \mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} \underbrace{\|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2}_{\text{condition}} + \underbrace{\mu\sigma_n^2 \|\mathbf{x} - \mathbf{z}_k\|^2}_{\text{consistence}} & \text{Data} \end{cases}$$

Plug-and-Play Image Restoration

$$\arg \min_{\mathbf{z}} \underbrace{\frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{z} - \mathbf{x}_k\|^2}_{\text{consistence}} + \underbrace{\mathcal{P}(\mathbf{z})}_{\text{prior}} \quad \Rightarrow \quad \mathbf{z}_k = \text{Denoiser}(\mathbf{x}_k)$$

degradation models

$$\arg \min_{\mathbf{x}} \underbrace{\|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2}_{\text{condition}} + \underbrace{\mu\sigma_n^2 \|\mathbf{x} - \mathbf{z}_k\|^2}_{\text{consistence}}$$

Inpainting $\mathbf{x}_{k-1} = \frac{\mathbf{M} \odot \mathbf{y} + \rho_k \mathbf{z}_k}{\mathbf{M} + \rho_k}$ Deblurring $\mathbf{x}_{k-1} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{y}) + \rho_k \mathcal{F}(\mathbf{z}_k)}{\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k}) + \rho_k} \right)$ SR $\mathbf{x}_{k-1} = \mathcal{F}^{-1} \left(\frac{1}{\rho_k} \left(\mathbf{d} - \overline{\mathcal{F}(\mathbf{k})} \odot_s \frac{(\mathcal{F}(\mathbf{k})\mathbf{d}) \Downarrow_s}{(\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k})) \Downarrow_s + \rho_k} \right) \right)$ $\mathbf{d} = \overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{y} \uparrow_{sf}) + \rho_t \mathcal{F}(\mathbf{z}_k)$	$\mathbf{y} = \mathbf{M} \odot \mathbf{x}$ $\mathbf{y} = \mathbf{x} \otimes \mathbf{k} + \mathbf{n}$ $\mathbf{y} = \mathbf{x} \downarrow_{sf}^{bicubic} + \mathbf{n}$
--	---

Approximately $\mathbf{x}_{k-1} \approx \mathbf{z}_k - \frac{1}{2\mu\sigma_n^2} \nabla_{\mathbf{z}_k} \|\mathbf{y} - \mathcal{H}(\mathbf{z}_k)\|^2$

Denoising Diffusion Models for Plug-and-Play Image Restoration

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}} \right)}_{\text{"predicted } x_0\text{"}} + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{"direction pointing to } x_t\text{"}}$$

One iteration HQS \rightarrow estimate $\mathbf{x}_0^{(t)}(\mathbf{x}_t, \mathbf{y})$

$$\begin{cases} \mathbf{z}_0^{(t)} = \arg \min_{\mathbf{z}} \frac{1}{2\bar{\sigma}_t^2} \|\mathbf{z} - \mathbf{x}_t\|^2 + \mathcal{P}(\mathbf{z}) \\ \mathbf{x}_0^{(t)} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \rho_t \|\mathbf{x} - \mathbf{z}_0^{(t)}\|^2 \end{cases}$$

Calculate the predicted conditional noise

$$\epsilon_\theta(\mathbf{x}_t, \mathbf{y}) = \frac{1}{\sqrt{1 - \bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0^{(t)}(\mathbf{x}_t, \mathbf{y}))$$

Finish one sampling step by adding noise back

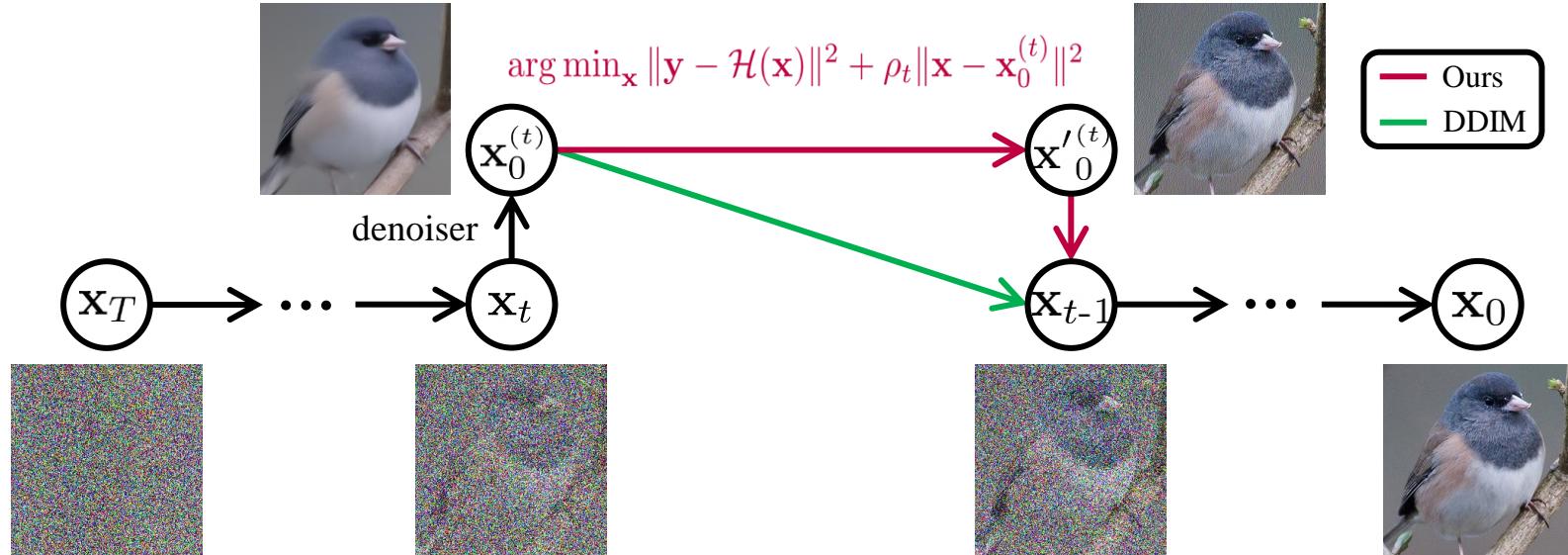
$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0^{(t)}(\mathbf{x}_t, \mathbf{y}) + \sqrt{1 - \bar{\alpha}_{t-1}} (\sqrt{1 - \zeta} \epsilon_\theta(\mathbf{x}_t, \mathbf{y}) + \sqrt{\zeta} \epsilon_t)$$

$$\hat{x}_0(x_t, y) = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t + (1 - \bar{\alpha}_t) \mathbf{s}_\theta(\mathbf{x}_t, y)) = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t + (1 - \bar{\alpha}_t) (\mathbf{s}_\theta(\mathbf{x}_t)) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y} | \mathbf{x}_t)) = \hat{x}_0(x_t) + \frac{1 - \bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \nabla_{\mathbf{x}} \log p_t(\mathbf{y} | \mathbf{x}_t)$$

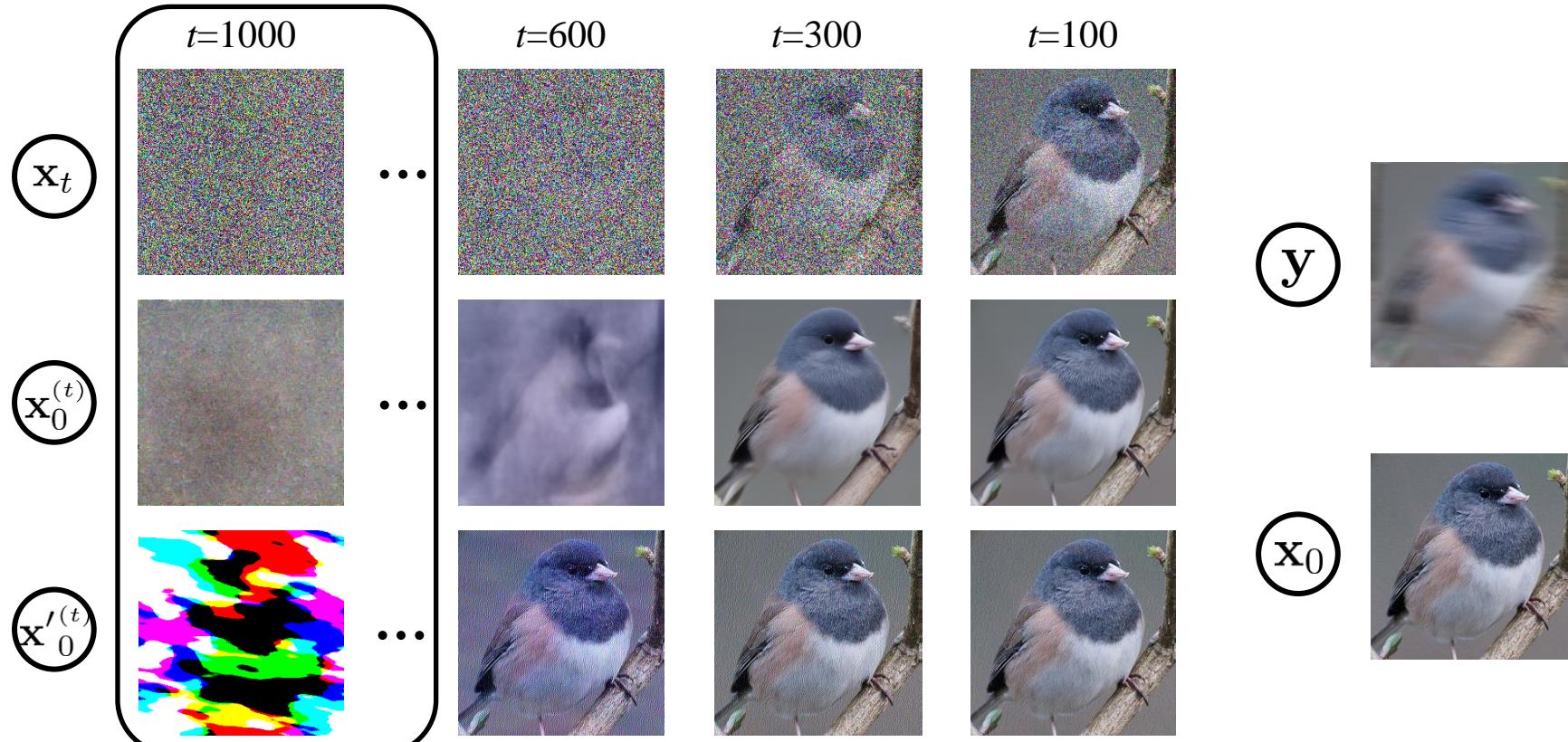
In this page we use ϵ_θ instead z_θ to avoid confusion

Denoising Diffusion Models for Plug-and-Play Image Restoration

$$p_{\theta}(x_{t-1}|x_t) \begin{cases} \hat{\mathbf{x}}_0^{(t)} = \frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \mathbf{z}_{\theta}^{(t)}(\mathbf{x}_t)}{\sqrt{\bar{\alpha}_t}} \\ \hat{\mathbf{x}}_0'^{(t)} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + (\lambda \sigma_n^2 / \bar{\sigma}_t^2) \|\mathbf{x} - \hat{\mathbf{x}}_0^{(t)}\|^2 \\ \mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \hat{\mathbf{x}}_0'^{(t)} + \sqrt{1 - \bar{\alpha}_{t-1}} (\sqrt{1 - \zeta} \hat{\mathbf{z}}_{\theta}^{(t)} + \sqrt{\zeta} \epsilon_t) \end{cases}$$

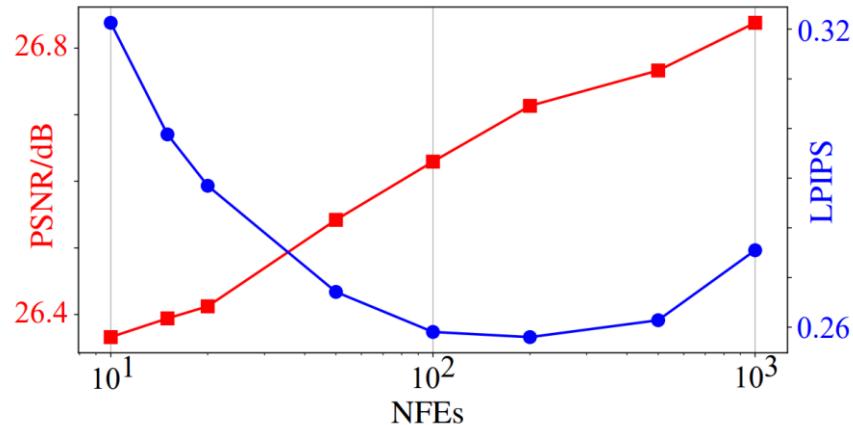


Denoising Diffusion Models for Plug-and-Play Image Restoration

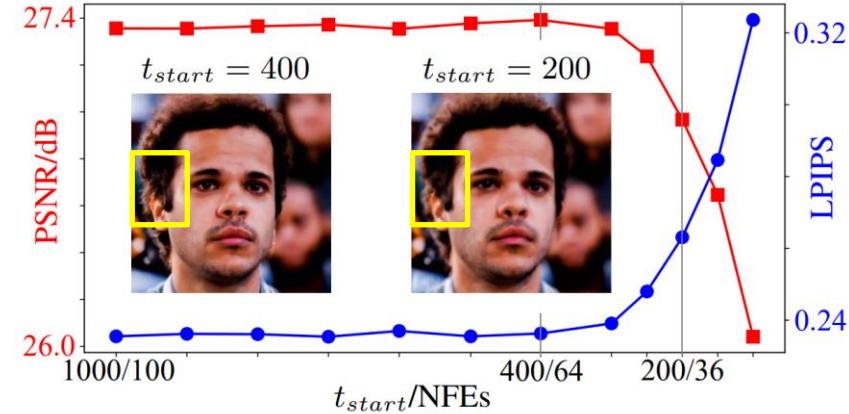


Ablation Study: Sampling Steps & Start Timestep

Effect of sampling steps

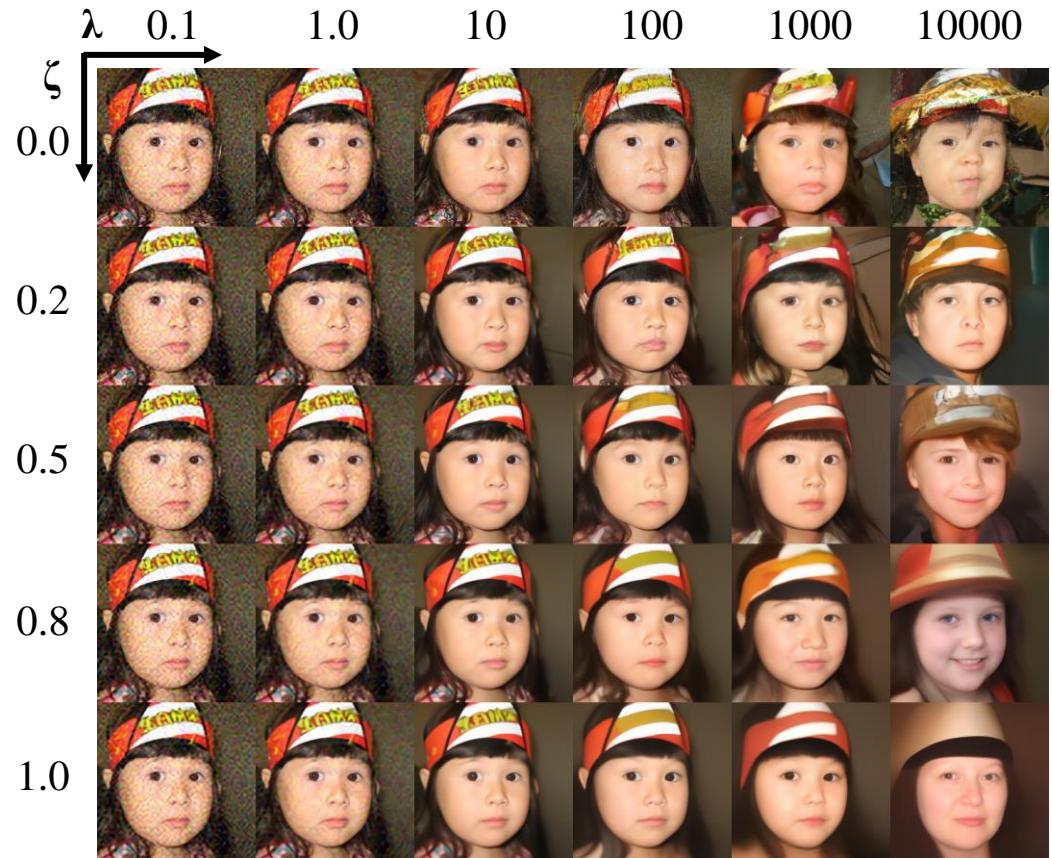


Effect of start sampling timestep



Ablation Study: Effect of Hyperparameters

- $\lambda < 1$ → the noise is amplified
- $\lambda > 1000 \rightarrow$ more *unconditional*
- $\zeta \sim 1 \rightarrow$ more blurry



Content

- Preliminaries
- Methods
- Results

Quantitative Results

FFHQ		Deblur (Gaussian)			Deblur (motion)			SR ($\times 4$)		
Method	NFEs ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
DiffPIR	100	27.36	59.65	0.236	26.57	65.78	0.255	26.64	65.77	0.260
DPS [8]	1000	25.46	65.57	0.247	23.31	73.31	0.289	25.77	67.01	0.256
DDRM [29]	20	25.93	101.89	0.298	-	-	-	27.92	89.43	0.265
DPIR [52]	>20	27.79	123.99	0.450	26.41	146.44	0.467	28.03	133.39	0.456
ImageNet		Deblur (Gaussian)			Deblur (motion)			SR ($\times 4$)		
Method	NFEs ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓
DiffPIR	100	22.80	93.36	0.355	24.01	124.63	0.366	23.18	106.32	0.371
DPS [8]	1000	19.58	138.80	0.434	17.75	184.45	0.491	22.16	114.93	0.383
DDRM [29]	20	22.33	160.73	0.427	-	-	-	23.89	118.55	0.358
DPIR [52]	>20	23.86	189.92	0.476	23.60	210.31	0.489	23.99	204.83	0.475

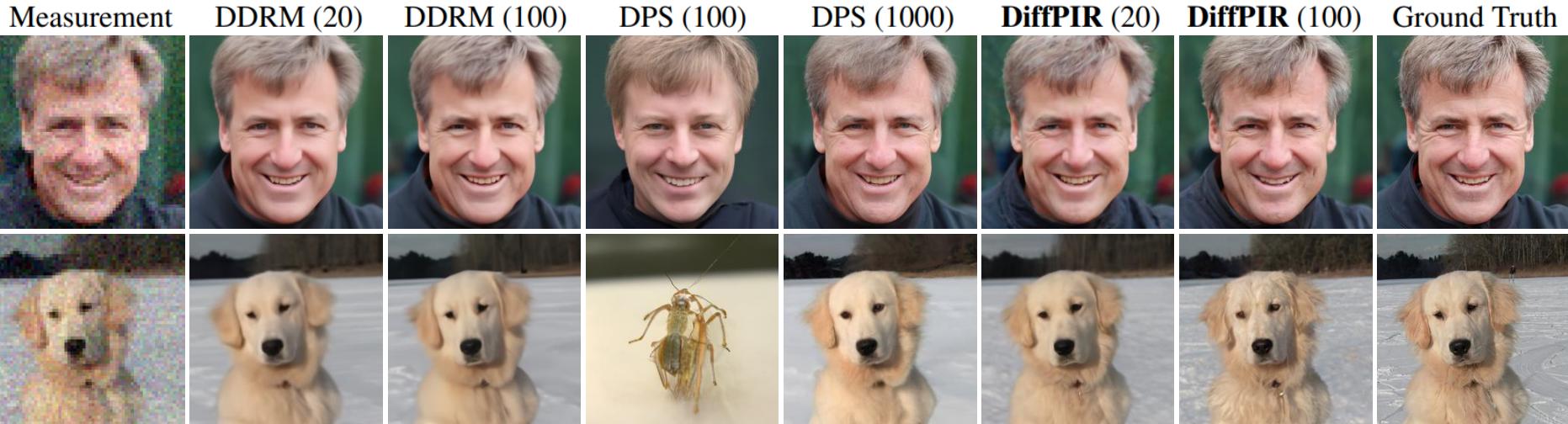
Table 1. **Noisy quantitative results on FFHQ (top) and ImageNet (bottom).** We compute the average PSNR (dB), FID and LPIPS of different methods on Gaussian deblurring, motion deblurring and 4× SR.

Quantitative Results

FFHQ		Inpaint (box)			Inpaint (random)			Deblur (Gaussian)			Deblur (motion)			SR ($\times 4$)		
Method	NFEs ↓	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS ↓	
DiffPIR	20	35.72	0.117	34.03	30.81	0.116	30.74	46.64	0.170	37.03	20.11	0.084	29.17	58.02	0.187	
DiffPIR	100	25.64	0.107	36.17	13.68	0.066	31.00	39.27	0.152	37.53	11.54	0.064	29.52	47.80	0.174	
DPS [8]	1000	43.49	0.145	34.65	33.14	0.105	27.31	51.23	0.192	26.73	58.63	0.222	27.64	59.06	0.209	
DDRM [29]	20	37.05	0.119	31.83	56.60	0.164	28.40	67.99	0.238	-	-	-	30.09	68.59	0.188	
DPIR [52]	>20	-	-	-	-	-	30.52	96.16	0.350	38.39	27.55	0.233	30.41	96.16	0.362	

Table 2. **Noiseless quantitative results on FFHQ.** We compute the average PSNR (dB), FID and LPIPS of different methods on inpainting, deblurring, and SR.

Qualitative Results: Noisy 4x SR



Qualitative Results: Noisy Motion Deblurring

Measurement



DPIR (20)



DPS (100)



DPS (1000)



DiffPIR (20)



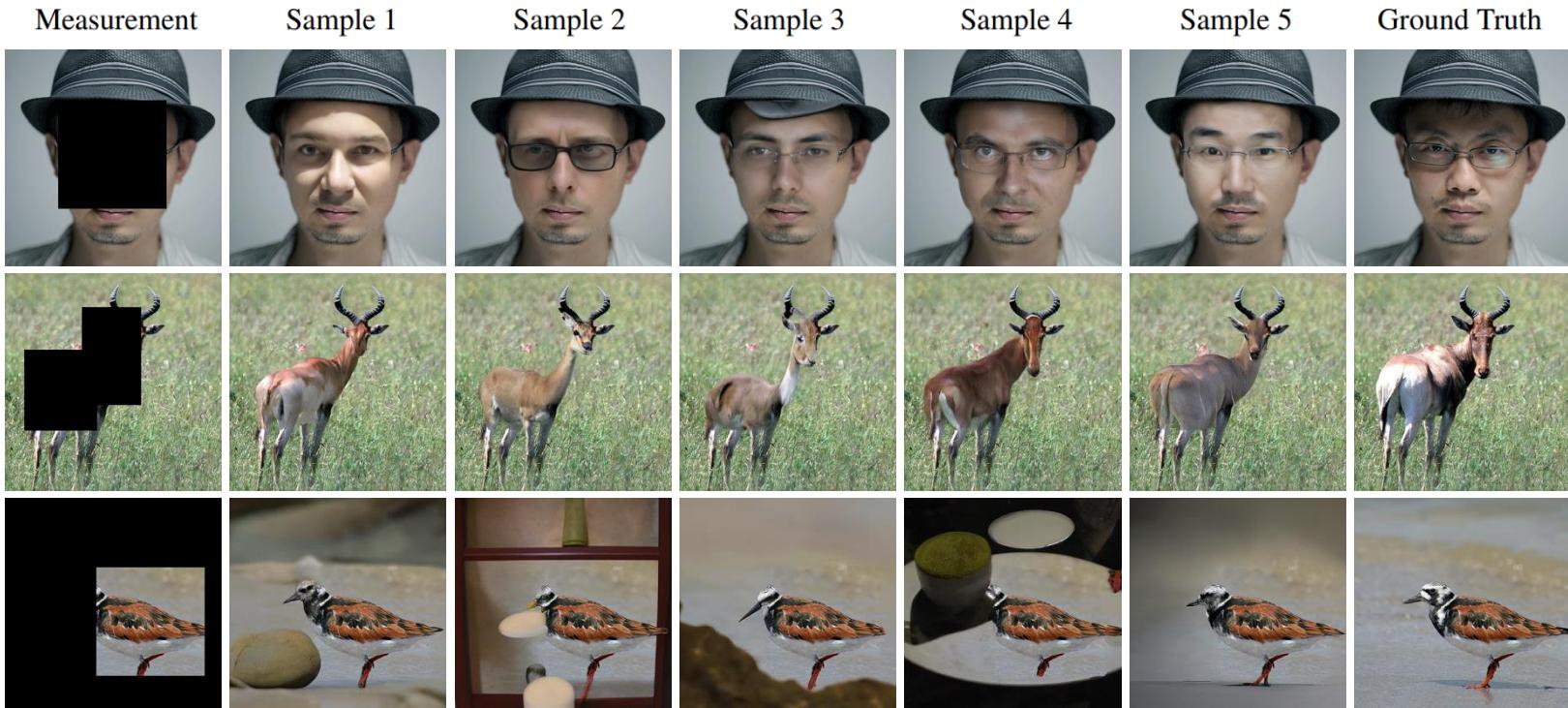
DiffPIR (100)



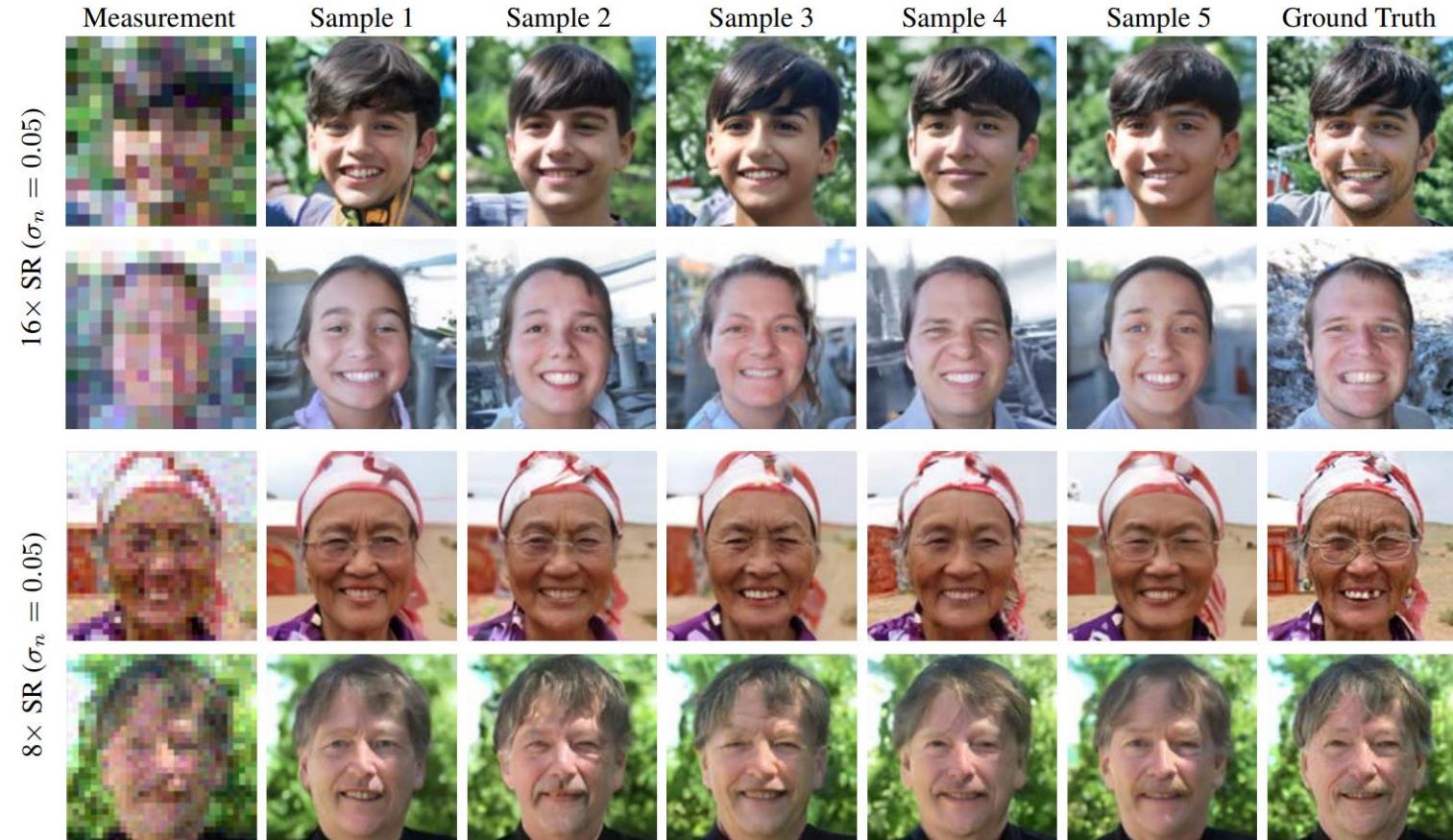
Ground Truth



Diverse Reconstruction: Inpainting



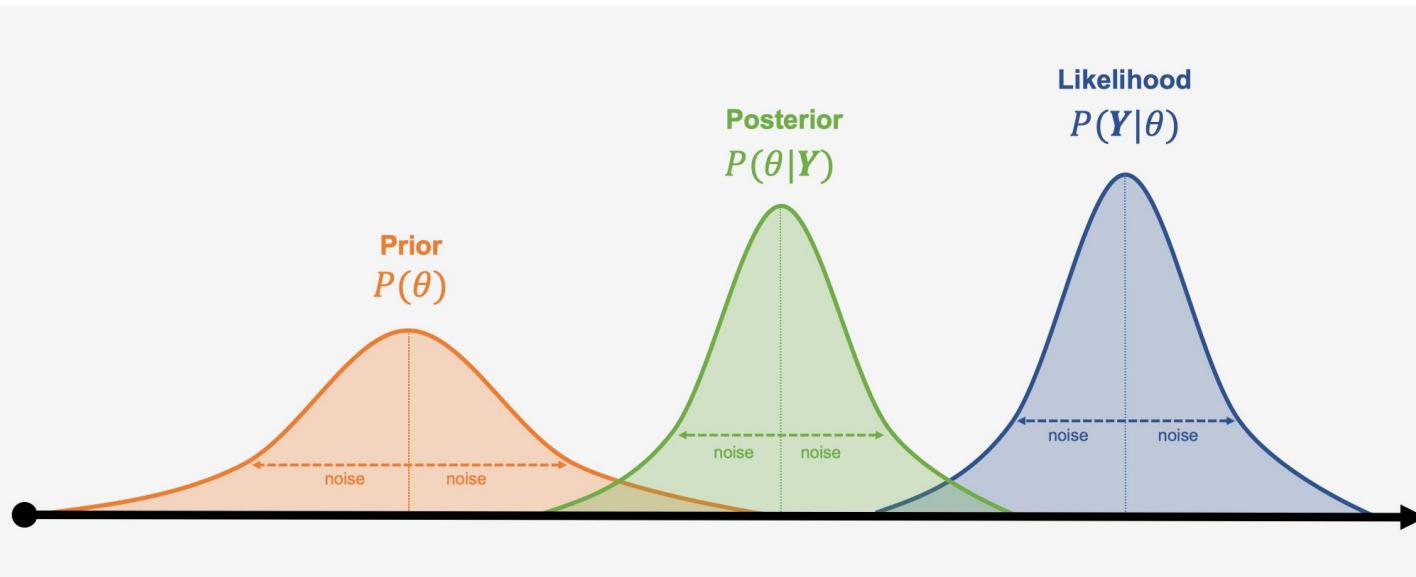
Diverse Reconstruction: Super Resolution



Thank You!

Additional Slides on Diffusion Models for IR

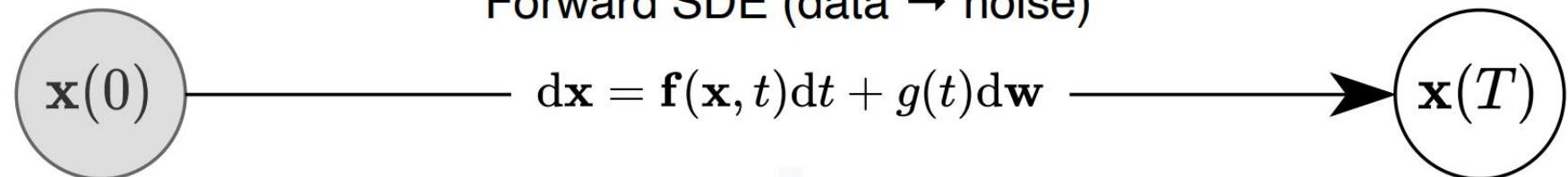
Sampling from the Posterior



REV. T. BAYES

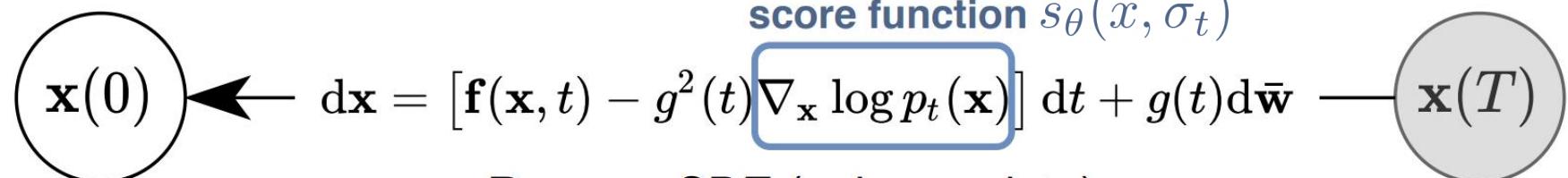
SDE-based Generative Models: A Unified Framework*

Forward SDE (data → noise)



score function $s_\theta(x, \sigma_t)$

Reverse SDE (noise → data)



SDE-based Generative Models: A Unified Framework

Training Objective (DSM)

$$\theta^* = \arg \min \mathbb{E}_{t \sim U(0, T)} \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)|x(0)} [\| s_\theta(x(t), t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0)) \|_2^2] \right\}$$

known Gaussian when $f(x, t)$ if affine

Discretizations

$$dx = f(x, t)dt + g(t)dw$$

SDE Form	Discrete Markov Chain	SDE Expression
Variance Exploding (VE) SDE (NCSN)	$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}$	$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw$
Variance Preserving (VP) SDE (DDPM)	$x_i = \sqrt{1 - \beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}$	$dx = \frac{1}{2} \beta(t) x dt + \sqrt{\beta(t)} dw$

SDE-based Generative Models: A Unified Framework

Model: DDPM and SDE point of views

Score in score-based model is affine transformation of predicted noise in DDPM

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\varepsilon} \quad \text{Equivalent one step forward}$$

$$\begin{aligned} s_\theta(\mathbf{x}_t, t) &\approx \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{x}_0) && \text{Denoising score matching} \\ &= -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0}{1 - \bar{\alpha}_t} && \text{Gaussian assumption} \\ &= -\frac{\boldsymbol{\varepsilon}}{\sqrt{1 - \bar{\alpha}_t}} && \leftarrow \\ &\approx -\frac{\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}} \end{aligned}$$

SDE-based Generative Models: A Unified Framework

Controllable Generation

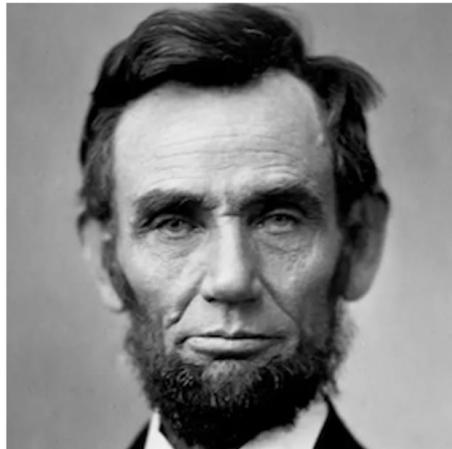
$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x|y)]dt + g(t)d\bar{w}$$

↓
Bayesian

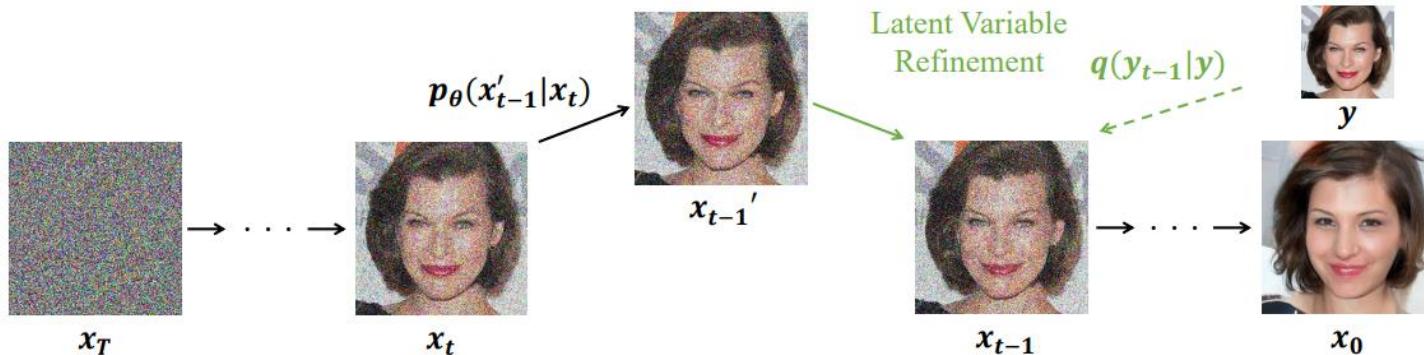
$$dx = \{f(x, t) - g(t)^2 [\nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)]\}dt + g(t)d\bar{w}$$

unconditional model

time-dependent classifier (?)



ILVR: Conditioning Method for DDPM*

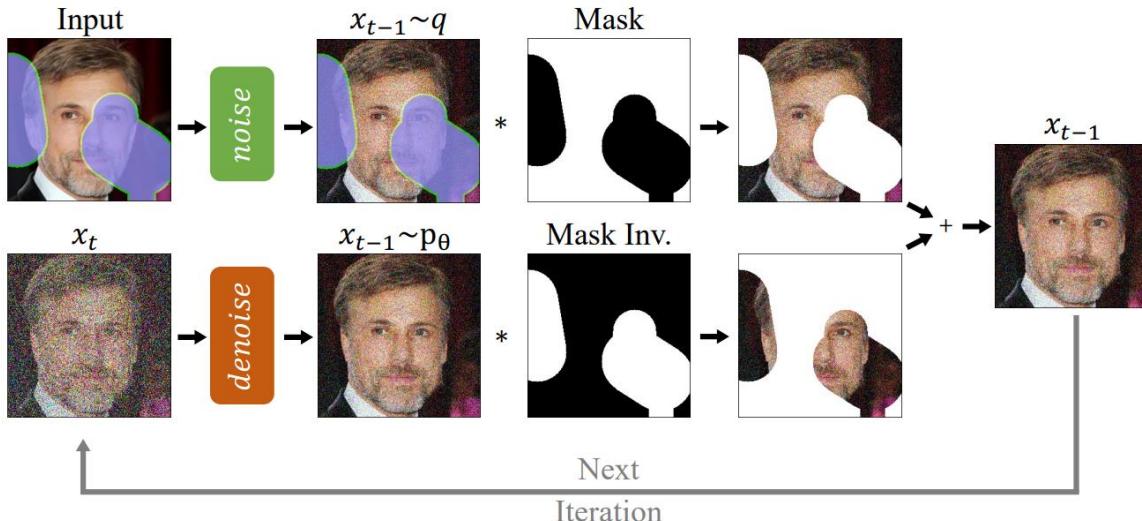


Algorithm 1 Iterative Latent Variable Refinement

```
1: Input: Reference image  $y$ 
2: Output: Generated image  $x$ 
3:  $\phi_N(\cdot)$ : low-pass filter with scale N
4: Sample  $x_T \sim N(\mathbf{0}, \mathbf{I})$ 
5: for  $t = T, \dots, 1$  do
6:    $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ 
7:    $x'_{t-1} \sim p_\theta(x'_{t-1} | x_t)$        $\triangleright$  unconditional proposal
8:    $y_{t-1} \sim q(y_{t-1} | y)$              $\triangleright$  condition encoding
9:    $x_{t-1} \leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})$  x_{t-1} = x'_{t-1} + a(\phi_N(y_{t-1}) - \phi_N(x'_{t-1}))
10: end for
11: return  $x_0$ 
```

RePaint: Inpainting using Denoising Diffusion Probabilistic Models*

Same idea but different downstream tasks from ILVR



Algorithm 1 Inpainting using our RePaint approach.

```
1:  $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:   for  $u = 1, \dots, U$  do
4:      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\epsilon = 0$ 
5:      $x_{t-1}^{\text{known}} = \sqrt{\alpha_t} x_0 + (1 - \bar{\alpha}_t) \epsilon$  unconditional
6:      $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $z = \mathbf{0}$ 
7:      $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$ 
8:      $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$ 
9:     if  $u < U$  and  $t > 1$  then
10:       $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}} x_{t-1}, \beta_{t-1} \mathbf{I})$ 
11:    end if
12:  end for
13: end for
14: return  $x_0$ 
```

Diffusion Posterior Sampling for General Noisy Inverse Problems*

general forward model $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}, \quad \mathbf{y}, \mathbf{n} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^d$

$$p(\mathbf{y}|\mathbf{x}_0) = \frac{1}{\sqrt{(2\pi)^n \sigma^{2n}}} \exp \left[-\frac{\|\mathbf{y} - \mathcal{A}(\mathbf{x}_0)\|_2^2}{2\sigma^2} \right]$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) \simeq s_{\theta^*}(\mathbf{x}_t, t) - \rho \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)\|_2^2$$

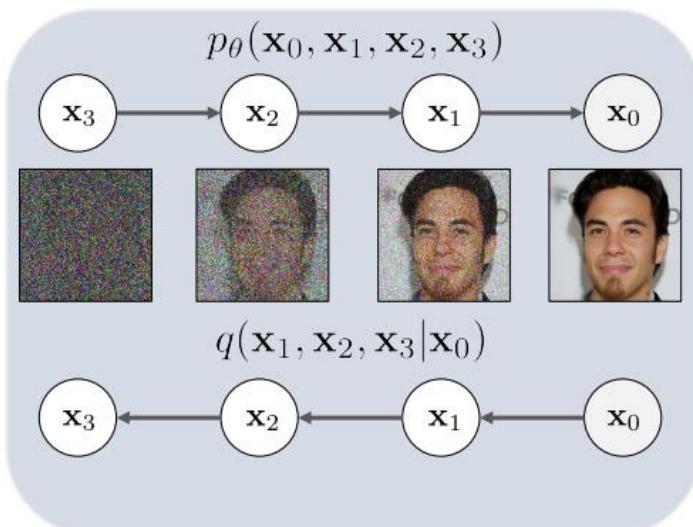
Algorithm 2 DPS - Gaussian [8]

Require: $N, y, \{\zeta_i\}_{i=1}^N, \{\tilde{\sigma}_i\}_{i=1}^N$

- 1: $x_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 2: **for** $i = N - 1$ **to** 0 **do**
 - 3: $\hat{s} \leftarrow s_\theta(x_i, i)$
 - 4: $\hat{x}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(x_i + \sqrt{1 - \bar{\alpha}_i} \hat{s})$
 - 5: $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 6: $x'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1 - \bar{\alpha}_{i-1})}{1 - \bar{\alpha}_i} x_i + \frac{\sqrt{\bar{\alpha}_{i-1}} \beta_i}{1 - \bar{\alpha}_i} \hat{x}_0 + \tilde{\sigma}_i z$
 - 7: $x_{i-1} \leftarrow x'_{i-1} - \zeta_i \nabla_{x_i} \|y - \mathcal{A}(\hat{x}_0)\|_2^2$
 - 8: **return** x_0
-

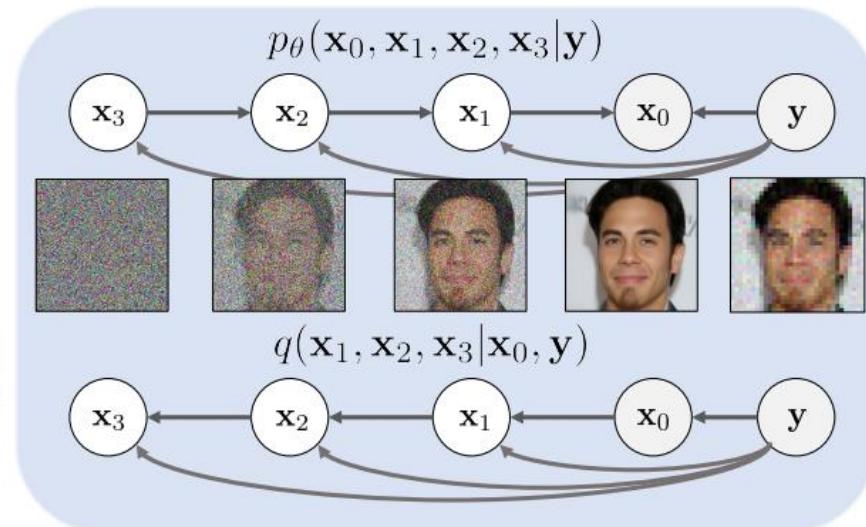
Denoising Diffusion Restoration Models (DDRM)*

An efficient, unsupervised posterior sampling method



Denoising Diffusion Probabilistic Models
(Independent of inverse problem)

Use pre-trained
models for linear
inverse problems



Denoising Diffusion Restoration Models
(Dependent on inverse problem)

Denoising Diffusion Restoration Models (DDRM)*

Linear inverse problem $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \iff \bar{\mathbf{y}} = \bar{\mathbf{x}}_0 + \bar{\mathbf{z}}$ $q(\bar{\mathbf{y}}^{(i)} | \mathbf{x}_0) = \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_y^2 / s_i^2)$

$$\text{SVD } \mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^\top \iff \bar{\mathbf{x}}_t = \mathbf{V}^T \mathbf{x}_t \\ \bar{\mathbf{y}} = \Sigma^\dagger \mathbf{U}^T \mathbf{y}$$

$q^{(T)}(\bar{\mathbf{x}}_T^{(i)} \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_y^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$ forward	$q^{(t)}(\bar{\mathbf{x}}_t^{(i)} \mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{t+1}}, \eta^2 \sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_y / s_i}, \eta^2 \sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_y}{s_i} \\ \mathcal{N}((1 - \eta_b) \bar{\mathbf{x}}_0^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_y^2}{s_i^2} \eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_y}{s_i} \end{cases}$ reverse
--	---

$$p_\theta^{(T)}(\bar{\mathbf{x}}_T^{(i)} | \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_y^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(0, \sigma_T^2) & \text{if } s_i = 0 \end{cases} \quad \text{singular values } s_1 \geq s_2 \geq \dots \geq s_m$$

$$\text{DDRM} \quad p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)} | \mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2 \sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_y / s_i}, \eta^2 \sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_y}{s_i} \\ \mathcal{N}((1 - \eta_b) \bar{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_y^2}{s_i^2} \eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_y}{s_i} \end{cases}$$

y null-space final steps generative part

Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems*

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) &\simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t) \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A}^T \left(\sigma^2 \mathbf{I} + \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \mathbf{A} \mathbf{A}^T \right)^{-1} \left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A} \mathbf{x}_t \right) \end{aligned}$$

A itself is row-orthogonal

$$[\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)]_m = \frac{\mathbf{a}_m^T \left(\mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A} \mathbf{x}_t \right)}{\sigma^2 \sqrt{\bar{\alpha}_t} + \frac{1 - \bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \|\mathbf{a}_m\|_2^2}$$

efficient computation via SVD

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) &\simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t) \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{V} \boldsymbol{\Sigma} \left(\sigma^2 \mathbf{I} + \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \boldsymbol{\Sigma}^2 \right)^{-1} \left(\mathbf{U}^T \mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \boldsymbol{\Sigma} \mathbf{V}^T \mathbf{x}_t \right), \end{aligned} \tag{12}$$

Algorithm 1: DMPS: DM based posterior sampling

Input: $\mathbf{y}, \mathbf{A}, \sigma^2, \{\tilde{\sigma}_t\}_{t=1}^T, \lambda$

Initialization: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

- 1 **for** $t = T$ **to** 1 **do**
- 2 Draw $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 3 $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{s}_{\theta}(\mathbf{x}_t, t) \right) + \tilde{\sigma}_t \mathbf{z}_t$
- 4 Compute $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$ as (12)
- 5 $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} + \lambda \frac{1 - \alpha_t}{\sqrt{\alpha_t}} \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$

Output: \mathbf{x}_0

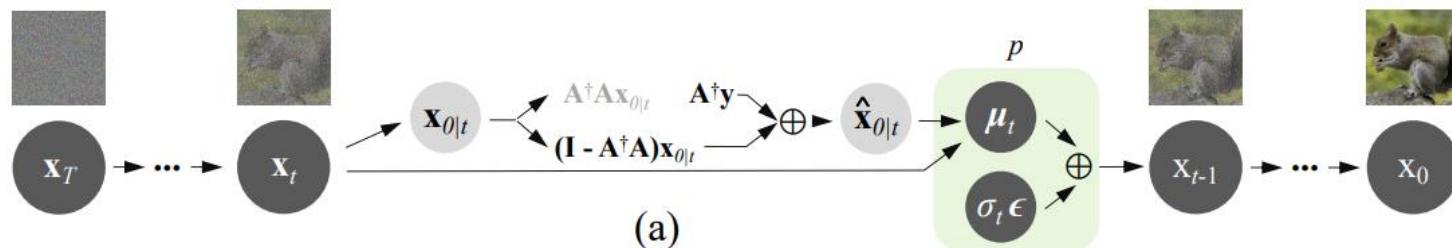
Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model*

Decouple $\mathbf{x} \equiv \underbrace{\mathbf{A}^\dagger \mathbf{A} \mathbf{x}}_{\text{range-space of } \mathbf{A}} + \underbrace{(\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}}_{\text{null-space of } \mathbf{A}}$

Consistency : $\mathbf{A}\hat{\mathbf{x}} \equiv \mathbf{y}$, *Realness* : $\hat{\mathbf{x}} \sim q(\mathbf{x})$

Reconstruction $\hat{\mathbf{x}} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \bar{\mathbf{x}}$ find a proper $\bar{\mathbf{x}}$ that makes the null-space term is in harmony with the range-space term

Diffusion Models $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}_{0|t}$



Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model*

Algorithm 1 Sampling of DDNM

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \mathcal{Z}_\theta(\mathbf{x}_t, t) \sqrt{1 - \bar{\alpha}_t})$ 
4:    $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}_{0|t}$ 
5:    $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$ 
6: return  $\mathbf{x}_0$ 

```

Algorithm 2 Sampling of DDNM⁺

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $L = \min\{T - t, l\}$ 
4:    $\mathbf{x}_{t+L} \sim q(\mathbf{x}_{t+L} | \mathbf{x}_t)$ 
5:   for  $j = L, \dots, 0$  do
6:      $\mathbf{x}_{0|t+j} = \frac{1}{\sqrt{\bar{\alpha}_{t+j}}} (\mathbf{x}_{t+j} - \mathcal{Z}_\theta(\mathbf{x}_{t+j}, t + j) \sqrt{1 - \bar{\alpha}_{t+j}})$ 
7:      $\hat{\mathbf{x}}_{0|t+j} = \mathbf{x}_{0|t+j} - \Sigma_{t+j} \mathbf{A}^\dagger (\mathbf{A} \mathbf{x}_{0|t+j} - \mathbf{y})$ 
8:      $\mathbf{x}_{t+j-1} \sim \hat{p}(\mathbf{x}_{t+j-1} | \mathbf{x}_{t+j}, \hat{\mathbf{x}}_{0|t+j})$ 
9: return  $\mathbf{x}_0$ 

```

